Claiming:

1. An interface for enabling communication between a primary and a secondary computing elements of a multiple element computer system, comprising:

a multiple element computing system having a primary computing element and a secondary computing element; and

a configuration and status register (CSR) block having a common section and a port specific section for enabling the exchange of buffer descriptors between said primary and said secondary computing elements, wherein said buffer descriptors contain information identifying underlying information packets to be processed on said computer system.

2. The invention in accordance with claim 1 wherein said buffer descriptors further comprise receive buffer descriptors sent from said secondary computing element to said primary computing element, and transmit buffer descriptors sent from said primary computing element to said secondary computing element.

3. The invention in accordance with claim 2 wherein said common section of said CSR block further comprises a queuing structure that said primary and said secondary computing elements use to receive and transmit said buffer descriptors.

4. The invention in accordance with claim 3 wherein said queuing structure further comprises:

a receive free pull queue wherein said secondary computing element can obtain an available receive buffer descriptor;

a receive queue having a head and a tail pointer, wherein said secondary computing element can return receive buffer descriptors to said tail of said receive queue after filling said receive buffer descriptors with data, and wherein said primary computing element can remove filled receive buffer descriptors from said head of said receive queue;

a receive free push queue wherein said primary computing element can return buffer descriptors after said primary computing element has finished processing said receive buffer descriptors;

a transmit free pull queue wherein said primary computing element can obtain an available transmit buffer descriptor;

a transmit queue having a head and a tail pointer, wherein said primary computing element can return transmit buffer descriptors to said tail of said transmit queue after filling said transmit buffer descriptors with data, and wherein said secondary computing element can remove filled transmit buffer descriptors from said head of said transmit queue;

a transmit done push queue wherein said secondary computing element can return transmit buffer descriptors after said secondary computing element has finished processing transmit buffer descriptors;

a transmit done pull queue wherein said primary computing element can retrieve transmit buffer descriptors; and

a transmit free push queue wherein primary computing element can return transmit buffer descriptors.

5. The invention in accordance with claim 1 wherein said common section of said CSR block further comprises an interrupt vector for communicating interrupts to said primary computing element.

6. The invention in accordance with claim 5 wherein said common section of said CSR block further comprises interrupt event and mask registers, wherein said interrupt event register will identify pending interrupts and said mask register will determine if said identified interrupts are sent to said primary computing element.

7. The invention in accordance with claim 1 wherein said common section of said CSR block further comprises a secondary computing element command register for sending commands to said secondary computing element.

8. The invention in accordance with claim 1 wherein said port specific section of said CSR block is Ethernet specific and comprises a physical address component and a transmitter and receiver control register to enable transmitting or receiving.

9. The invention in accordance with claim 1 wherein said port specific section of said CSR block is ATM specific and comprises a physical address component and a transmitter and receiver control register to enable transmitting or receiving.

10. The invention in accordance with claim 2 wherein said receive buffer descriptors comprise:

a next buffer descriptor field for storing a pointer to a next buffer descriptor in a chain of buffer descriptors;

a control/status field comprising information for distinguishing between receive and transmit buffer descriptors, and if said receive buffer descriptor is Ethernet specific comprising information about said underlying information packets, or if said receive buffer descriptor is ATM specific an interrupt bit causing said secondary computing element to send an interrupt to said primary computing element;

a port number field comprising network and interface specific information;

a data length field comprising the number of bytes containing said underlying information packet read into a buffer;

a data size field comprising the maximum number of bytes that can be read into said buffer;

a data location field comprising a pointer to the beginning of said buffer; and

a user defined data field that is not modified by said secondary computing element.

11. The invention in accordance with claim 2 wherein said transmit buffer descriptors comprise:

a next buffer descriptor field for storing a pointer to a next buffer descriptor in a chain of buffer descriptors;

a control/status field comprising information for distinguishing between receive and transmit buffer descriptors, and an interrupt bit causing the secondary computing element to send an interrupt to said primary computing element;

a port number field comprising network and interface specific information;

a data length field comprising the number of bytes containing said underlying information packet read into a buffer;

a data location field comprising a pointer to the beginning of said buffer; and

a user defined data field that is not modified by said secondary computing element.

12. The invention in accordance with claim 1 wherein said computer system comprises a network processor with a core processor and at least one microengine, and said primary computing element is said core processor and said secondary computing element is said microengine.

13. The invention in accordance with claim 12 further comprising:

a forwarding table manager application running on said primary and said secondary computing elements wherein said forwarding table manager:

builds a table comprised of a plurality of entries with addresses associated therewith wherein said entries are organized hierarchically according to an LC-Trie compression algorithm operating on said addresses;

receives an information packet within said computer system wherein said information packet has a destination address associated therewith;

searches said table using an LC-Trie search algorithm to find a match between said address of an entry in said table and said destination address of said information packet; and

transmits said information packet to a forwarding address associated with said address of said matching entry.

14. The invention in accordance with claim 13 wherein said table comprises an LC-Trie search table and a next-hop table associated together, wherein said LC-Trie search table comprises information from said LC-Trie compression algorithm, and wherein said next-hop table comprises information necessary to transmit said information packet to said forwarding address associated with said matching entry.

15. The invention in accordance with claim 14 wherein said LC-Trie search table entries comprise a branching factor, a skip value, and an LC-Trie/Next-Hop Offset generated for each of said plurality of entries by said LC-Trie compression algorithm when said LC-Trie search table was built.

16. The invention in accordance with claim 15 wherein said next-hop table entries comprise an address field containing said address of said matching entry identified in said

searching step, and an opaque data field for storing specialized packet processing information.

17. The invention in accordance with claim 16 wherein said next-hop table entries further comprise a mask length field containing a mask length of said entry, and said forwarding table manager application verifies that said address of said matching entry and said destination address of said information packet match to at least said mask length.

18. The invention in accordance with claim 17 wherein said next-hop table entries further comprise a next-hop backup offset field that references a previous entry in the hierarchy created in the building step, and said forwarding table manager application performs a second verification performed if said first verification fails, which verifies that said address of said previous entry and said destination address of said information packet match to at least a mask length number of bits of said previous entry.

19. The invention in accordance with claim 16 wherein said opaque data field further comprises MPLS tags.

20. The invention in accordance with claim 16 wherein said opaque data field further comprises quality of service parameters.

21. The invention in accordance with claim 16 wherein said opaque data field further comprises encryption handling parameters.

22. The invention in accordance with claim 16 wherein said addresses comprise IP addresses, and said opaque data field further comprises VLAN tags.

23. The invention in accordance with claim 16 wherein said opaque data field further comprises a port specific field for accessing said forwarding address identified in said transmitting step.

24. The invention in accordance with claim 23 wherein said addresses are IP addresses.

25. The invention in accordance with claim 18 wherein said next-hop table entries further comprise a flag field wherein if said flag is set said port specific field contains an offset to an entry in said next hop table containing said forwarding IP address indicating said forwarding IP address addresses a network route, and if the flag is not set said port specific field contains said forwarding IP address indicating said forwarding IP address addresses a host route.

26. The invention in accordance with claim 13 wherein said forwarding table manager application builds said table utilizing said core processor, and said search of said table is performed by said forwarding table manager application utilizing said microengine.

27. An interface for enabling communication between a core processor and at least one microengine of a network processor, comprising:

a network processor having a core processor and at least one microengine;

a configuration and status register (CSR) block having a common section and a port specific section for enabling the exchange of buffer descriptors between said core processor and said microengine, wherein said buffer descriptors contain information identifying underlying information packets;

wherein said buffer descriptors further comprise receive buffer descriptors sent from said microengine to said core processor, and transmit buffer descriptors sent from said core processor to said microengine;

wherein said common section of said CSR block further comprises:

a queuing structure that said core processor and said microengine use to receive and transmit said buffer descriptors, comprising:

a receive free pull queue wherein said microengine can obtain an available receive buffer descriptor;

a receive queue having a head and a tail pointer, wherein said microengine can return receive buffer descriptors to said tail of said receive queue after filling said receive buffer descriptors with data, and wherein said core processor can remove filled receive buffer descriptors from said head of said receive queue;

a receive free push queue wherein said core processor can return buffer descriptors after said core processor has finished processing said receive buffer descriptors;

a transmit free pull queue wherein said core processor can obtain an available transmit buffer descriptor;

a transmit queue having a head and a tail pointer, wherein said core processor can return transmit buffer descriptors to said tail of said transmit queue after filling said transmit buffer descriptors with data, and wherein said microengine can remove filled transmit buffer descriptors from said head of said transmit queue;

a transmit done push queue wherein said microengine can return transmit buffer descriptors after said microengine has finished processing transmit buffer descriptors;

a transmit done pull queue wherein said core processor can retrieve transmit buffer descriptors; and

a transmit free push queue wherein core processor can return transmit buffer descriptors;

an interrupt vector for communicating interrupts to said core processor;

interrupt event and mask registers, wherein said interrupt event register will identify pending interrupts and said mask register will determine if said identified interrupts are sent to said core processor;

a microengine command register for sending commands to said microengine;

wherein said port specific section of said CSR block comprises an Ethernet specific component comprising a physical address component and a transmitter and receiver control register to enable transmitting or receiving, and comprises an ATM specific

component comprising a physical address component and a transmitter and receiver control register to enable transmitting or receiving;

wherein said receive buffer descriptors comprise:

a next buffer descriptor field for storing a pointer to a next buffer descriptor in a chain of buffer descriptors;

a control/status field comprising information for distinguishing between receive and transmit buffer descriptors, and if said receive buffer descriptor is Ethernet specific comprising information about said underlying information packets, or if said receive buffer descriptor is ATM specific an interrupt bit causing said microengine to send an interrupt to said core processor;

a port number field comprising network and interface specific information;

a data length field comprising the number of bytes containing said underlying information packet read into a buffer;

a data size field comprising the maximum number of bytes that can be read into said buffer;

a data location field comprising a pointer to the beginning of said buffer; and

a user defined data field that is not modified by said microengines; and

wherein said transmit buffer descriptors comprise:

a next buffer descriptor field for storing a pointer to a next buffer descriptor in a chain of buffer descriptors;

a control/status field comprising information for distinguishing between receive and transmit buffer descriptors, and an interrupt bit causing the microengine to send an interrupt to said core processor;

a port number field comprising network and interface specific information;

a data length field comprising the number of bytes containing said underlying information packet read into a buffer;

a data location field comprising a pointer to the beginning of said buffer; and

a user defined data field that is not modified the microengine.